

**ВЫСОКОУРОВНЕВЫЕ МЕТОДЫ  
ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

*Руководство по выполнению курсовых работ*

**Владивосток  
2006**

Министерство образования и науки Российской Федерации  
Владивостокский государственный университет экономики и сервиса

## **ВЫСОКОУРОВНЕВЫЕ МЕТОДЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

*Руководство по выполнению курсовых работ*  
по специальности  
351400 Прикладная информатика (в экономике)

Владивосток  
Издательство ВГУЭС  
2006

ББК ■.■

Руководство по выполнению курсовых работ по дисциплине "Высокоуровневые методы информатики и программирования" составлено в соответствии с требованиями государственного стандарта России. Предназначено для студентов специальности 351400 "Прикладная информатика (в экономике)".

Составитель: Люлько В.И., ст. преподаватель кафедры информатики, инженерной и компьютерной графики.

Утверждено на заседании кафедры информатики, инженерной и компьютерной графики 12.05.2006 г., протокол № 18

Рекомендовано к изданию методическим советом института информационных технологий и технических систем ВГУЭС ■.■.2006 г., протокол № ■

© Издательство Владивостокского государственного университета экономики и сервиса, 2006

## ВВЕДЕНИЕ

Профессиональная деятельность по специальности "Прикладная информатика (в экономике)" предполагает как создание, внедрение, анализ, сопровождение и эксплуатацию профессионально-ориентированных информационных систем, связанных с обработкой экономической информации, так и разработку и эксплуатацию отдельных самостоятельно используемых программ.

Выполнение курсовых работ в рамках изучения дисциплины "Высокоуровневые методы информатики и программирования" обеспечивает получение практических навыков использования современных средств разработки программного обеспечения с применением технологии визуального программирования и, как следствие, повышение уровня практической подготовленности специалистов, необходимого для успешного применения специальных программных средств при проектировании, создании и использовании профессионально-ориентированных пользовательских оболочек информационных систем и отдельных программ.

В предлагаемом руководстве содержатся перечень тем курсовых работ, методические указания по их выполнению и список рекомендуемой литературы.

При подготовке руководства принималось во внимание, что студенты указанной выше специальности "Прикладная информатика (в экономике)" в период выполнения курсовой работы изучают материал по курсу "Высокоуровневые методы информатики и программирования" в соответствии с темами лекционных и лабораторных занятий этого курса (1), а также уже имеют опыт выполнения курсовых работ по дисциплине "Информатика и программирование", см., например, (8). В связи с этим ряд общих вопросов, имеющих отношение к использованию компонентов и технике работы в средах программирования как по дисциплине "Информатика и программирование", так и по дисциплине "Высокоуровневые методы информатики и программирования", в настоящем руководстве могут подробно не рассматриваться.

Содержание значительной части разделов и подразделов настоящего руководства не связано с конкретной средой программирования. Заголовки разделов и подразделов, в которых упоминается конкретная среда программирования, отмечены символом \*.

# **1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

## **1.1 Цель и задачи курсовой работы**

Целью выполнения курсовой работы является практическое использование полученных базовых знаний о современных методах проектирования и разработки прикладных программ и информационных систем на основе концепций объектно-ориентированного подхода к проектированию и разработке программ, а также практическое освоение современных технологий программирования и концепций создания программных приложений, развитие навыков самостоятельной работы, связанных с анализом, детализацией, выбором методов решения поставленных задач, планированием использования возможностей сред программирования, а также различных источников информации для реализации задач курсовой работы.

В ходе выполнения курсовой работы по данной дисциплине у студента должно расширяться и углубляться представление о принципах и возможностях использования инструментальных систем программирования.

В ходе достижения цели решаются следующие задачи:

дальнейшего развития логического и алгоритмического мышления; углубленного изучения принципов работы прикладного программного обеспечения;

выработки умения использовать современные инструментальные средства для разработки, отладки и тестирования создаваемых прикладных программ;

выработки умения самостоятельного анализа, детализации, выбора методов решения задач, планированием использования возможностей сред программирования, а также различных источников информации для реализации этих задач.

## **1.2 Требования к выполнению курсовой работы**

При выполнении курсовой работы студент должен ознакомиться с предметной областью по теме выполняемой работы и в необходимой мере исследовать эту область.

В соответствии с темой курсовой работы студент должен отобрать и проанализировать информацию, необходимую для реализации задач курсовой работы, и выбрать способы решения этих задач с использованием возможностей соответствующих сред программирования.

### **1.3 Объем и содержание курсовой работы**

Выполненная курсовая работа должна состоять из текстовой части работы и одного или нескольких проектов пользовательских приложений, созданных в заданных средах программирования, или только в виде текстовой части работы.

Вид представления курсовой работы (наличие текстовой части с проектами или только текстовой части) определяется, прежде всего, темой работы, но может быть изменен при получении студентом задания на выполнение курсовой работы или в процессе ее выполнения.

Объемы текстовой части работы и других файлов заранее никак не ограничиваются, но могут быть ограничены при получении студентом задания на выполнение курсовой работы или в процессе ее выполнения.

### **1.4 Техническое и программное обеспечение дисциплины**

Для выполнения курсовой работы рекомендуется использовать персональные компьютеры: Pentium III с тактовой частотой процессора 500 МГц и выше, оперативной памятью 128 Мбайт и более, памятью на жестком диске 10 Гбайт и более; программное обеспечение: операционная система Windows 2000 и выше, выбранная система программирования, пакет Microsoft Office 2000 и выше.

### **1.5 Виды контроля знаний студентов и их отчетности по курсовой работе**

Выполненная курсовая работа должна быть проверена преподавателем и, после устранения его замечаний, защищена студентом. При использовании дистанционных технологий обучения курсовая работа оценивается без защиты.

Итогом выполнения курсовой работы по дисциплине "Высокоуровневые методы информатики и программирования" является зачет с дифференцированными оценками, который проводится: для студентов очной формы обучения – на зачетной неделе, для студентов заочной формы обучения – в период экзаменационно-лабораторной сессии до экзамена по указанной дисциплине, при использовании дистанционных технологий обучения – до защиты дипломной работы.

Соответствующие зачетные требования для студентов очной формы обучения определяются, кроме того, в соответствии с действующей во ВГУЭС рейтинговой системой оценки успеваемости студентов.

## 2. ТЕМАТИКА КУРСОВЫХ РАБОТ

Выполнение курсовых работ по дисциплине "Высокоуровневые методы информатики и программирования" может быть связано:

а) с реализацией возможностей, предоставляемых средами программирования, в виде программных приложений, ориентированных на конечного пользователя (см. ниже темы 1-16 примерного перечня тем курсовых работ);

б) с изучением возможностей использования компонентов сред программирования и реализацией этих возможностей в виде программных приложений демонстрационного характера (см. там же темы 17-25);

в) с изучением информационных и образовательных ресурсов, имеющих отношение к использованию сред программирования (см. там же темы 26-28).

Примерный перечень тем курсовых работ:

1. Программа: Решение алгебраических уравнений 1-й, 2-й, 3-й и 4-й степени.
2. Программа: Решение системы  $N$  линейных алгебраических уравнений с  $N$  неизвестными.
3. Программа: Операции с алгебраическими матрицами.
4. Программа: Двоичный калькулятор.
5. Программа: Восьмеричный калькулятор.
6. Программа: Шестнадцатеричный калькулятор.
7. Программа: Электронный кассовый аппарат.
8. Программа: Переводы единиц измерения.
9. Программа: Оболочка для тестирования
10. Программа: Оболочка для анкетирования.
11. Программа: Календарь с часами.
12. Программа: Многофункциональные электронные часы.
13. Программа: Многофункциональный секундомер.
14. Программа: Протокол соревнований.
15. Программа: Таблица чемпионата.
16. Программа: Поиск сбойных участков на диске.
17. Форма, ее свойства и примеры использования.
18. Возможности и примеры использования окон редактирования и меток.
19. Возможности и примеры использования кнопок, флажков, переключателей.
20. Возможности и примеры использования списков.
21. Возможности и примеры использования панелей.
22. Изобразительные возможности среды программирования.
23. Возможности среды программирования для работы с файлами.

24. Сравнение возможностей сред программирования на примере создания и использования простого приложения.
25. Сравнение возможностей сред программирования на примерах использования компонентов.
26. Обзор литературы и CD по среде программирования.
27. Internet-поддержка среды программирования.
28. Образовательные Internet-ресурсы дисциплины "Высокоуровневые методы информатики и программирования".

### **3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВЫХ РАБОТ**

#### **3.1 Выбор темы курсовой работы и получение задания на ее выполнение**

Студент выбирает тему курсовой работы по согласованию с преподавателем.

После выбора темы работы студент должен получить от преподавателя конкретное задание на выполнение работы по выбранной теме.

Это задание может включать в себя требования и рекомендации по использованию входных и выходных данных, форм и компонентов, представлению результатов выполнения курсовой работы, график реализации и представления результатов выполнения отдельных частей работы и т.п.. В процессе выполнения курсовой работы детали реализации задания могут быть уточнены.

Кроме того, в задание на выполнение курсовой работы могут быть включены указания и рекомендации по ее выполнению (использование тех или иных способов и приемов реализации задания, рекомендуемые источники информации и т.п.). В процессе выполнения курсовой работы эти указания и рекомендации могут дополняться и уточняться.

Тема курсовой работы может быть заменена преподавателем по просьбе студента при наличии независящих от студента объективных обстоятельств, препятствующих выполнению задания по выбранной теме. При замене темы по неуважительным причинам оценка выполненной работы по замененной теме снижается.

При дистанционных технологиях обучения и, в исключительных случаях, для студентов заочной формы обучения тематика курсовых работ, а также методические указания по их самостоятельному выполнению могут быть включены в состав соответствующих электронных учебных материалов и предоставлены студентам с использованием электронных средств связи.

При дистанционных технологиях обучения и, в исключительных случаях, для студентов заочной формы обучения выбор темы курсовой работы и получение задания на ее выполнение могут производиться с использованием электронных средств связи.

### 3.2 Порядок размещения и рекомендуемые имена папок и файлов с результатами работы

При выполнении курсовой работы должен быть создан файл электронного варианта текстовой части работы. Для большинства курсовых работ также могут быть созданы файлы проектов приложений. Кроме того, могут быть созданы файлы с другой информацией, не включенной в текстовую часть работы.

При выполнении курсовой работы рекомендуется все упомянутые выше файлы располагать на жестком диске компьютера в папке, имя которой должно начинаться с номера группы и фамилии студента и, при необходимости, других идентифицирующих признаков.

В этой папке должен находиться файл с текстовой частью курсовой работы и папка с проектами. Имя файла с текстовой частью работы должно совпадать с фамилией студента.

Папка с файлами проектов должна иметь имя Проекты (при создании одного проекта эта папка должна иметь имя Проект). При создании нескольких проектов в папке с проектами должны находиться вложенные папки с файлами соответствующих проектов. Имя каждой из этих вложенных папок должно соответствовать назначению и выражать краткую суть соответствующего проекта (см. пример на рисунке 1).

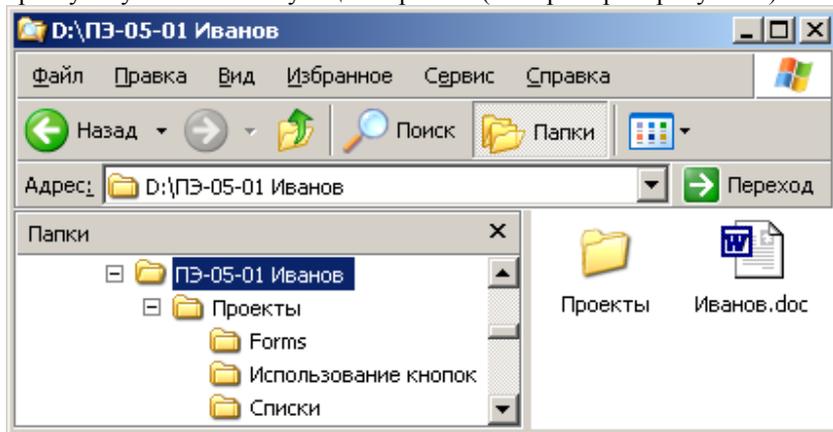


Рис. 1

Имена файлов проектов также должны соответствовать назначению и выражать краткую суть соответствующего проекта в рамках

ограничений, накладываемых средой программирования на имена этих файлов.

Не рекомендуется использовать стандартные имена, присваиваемые папкам и файлам по умолчанию операционной системой или средой программирования, а также имена общего характера (типа Папка1, Файл2 и т.п.) и жаргонные имена.

При сохранении файлов каждого проекта следует создавать и использовать только одну и ту же папку для размещения файлов этого проекта. Эту папку можно сразу же создать внутри упомянутой выше папки Проекты (при создании и сохранении файлов одного проекта следует использовать папку Проект).

Все приведенные в данном руководстве дальнейшие указания, связанные с использованием папок и файлов в курсовой работе, будут предполагать следование приведенным выше рекомендациям.

Иной порядок размещения и иные имена папок и файлов электронного варианта курсовой работы могут быть определены при получении студентом задания на выполнение курсовой работы или в процессе ее выполнения.

В процессе выполнения курсовой работы настоятельно рекомендуется периодически копировать промежуточные результаты выполнения работы, сохраняя эти результаты на дискетах, Flash-накопителях и других носителях информации на случай утраты возможности доступа к результатам, находящимся на компьютере. Для уменьшения объема занимаемой дисковой памяти соответствующие папки и файлы перед копированием могут быть заархивированы.

### **3.3 Выполнение курсовой работы**

При выполнении курсовой работы студент имеет право получать консультации преподавателя по вопросам, связанным с выполнением этой работы.

Преподаватель имеет право отказать студенту в получении консультации и предложить ему самостоятельно разобраться с возникшими вопросами, если ответы на эти вопросы не выходят за рамки материала, который должен уже знать студент по курсу дисциплины "Высокоуровневые методы информатики и программирования".

Студент обязан выполнять установленный преподавателем график реализации и представления результатов выполнения отдельных частей работы.

Студент имеет право до начала зачетной недели (для заочной формы обучения – до начала экзаменационно-лабораторной сессии) сдать выполненную в полном объеме курсовую работу в электронном варианте на проверку преподавателю и получить ее на доработку с

замечаниями преподавателя для устранения этих замечаний без снижения окончательной оценки за выполненную работу.

При дистанционных технологиях обучения и для студентов заочной формы обучения все вопросы, возникающие на этапе выполнения курсовой работы, могут быть решены с использованием электронных средств связи.

### **3.4 Представление результатов выполнения курсовой работы**

Выполненная курсовая работа может быть представлена в полном объеме в электронном варианте на проверку преподавателю и, после устранения замечаний, представлена в окончательном виде с текстовой частью в бумажном варианте.

Полностью выполненная курсовая работа в окончательном виде должна быть представлена в виде текстовой части в бумажном варианте, к которой должны прилагаться машинные носители информации (одна или несколько дискет, компакт-диск и т.п.) с электронным вариантом курсовой работы, включающим в себя файл электронного варианта текстовой части, а также, для большинства курсовых работ, файлы проектов приложений и файлы с другой информацией, не включенной в бумажный вариант текстовой части работы, или вообще не включенные в текстовую часть работы.

Для уменьшения объема занимаемой дисковой памяти папка с файлами курсовой работы может быть заархивирована в виде одного архивного файла с именем, совпадающим с именем этой папки.

При невозможности архивации сразу всей папки с курсовой работой файлы или группы файлов в папках могут быть заархивированы при условии сохранения структуры папок и присвоении получившимся архивным файлам имен, совпадающих с именами соответствующих заархивированных файлов (при архивации отдельных файлов), или имен, совпадающих с именами папок (при архивации в виде одного архивного файла всех файлов, находящихся в папке).

Перед архивацией из числа упомянутых выше файлов проектов следует исключить выполняемые программные файлы (занимающие много дисковой памяти) и резервные файлы.

Иные варианты представления курсовой работы могут быть определены при получении студентом задания на выполнение курсовой работы или в процессе ее выполнения.

При использовании дистанционных технологий обучения бумажный вариант не представляется. Электронный вариант курсовой работы должен быть заархивирован желательно в виде одного архивного файла, и затем этот файл может быть передан по

электронным средствам связи для проверки преподавателю. При использовании нескольких архивных файлов имена этих файлов должны отличаться друг от друга (имя каждого из этих файлов должно начинаться с номера группы и фамилии студента и содержать присущие только имени этого файла идентифицирующие признаки).

При использовании дистанционных технологий обучения и для студентов заочной формы обучения все вопросы, связанные с представлением результатов выполнения курсовой работы, могут быть решены с использованием электронных средств связи.

### **3.5 Сроки выполнения курсовой работы**

Курсовая работа должна быть выполнена студентом и сдана в полном объеме в электронном варианте на проверку преподавателю до начала зачетной недели (для заочной формы обучения – в начале экзаменационно-лабораторной сессии).

Проверенный электронный вариант сданной курсовой работы может быть возвращен студенту с замечаниями преподавателя на доработку до срока, установленного преподавателем для окончательной сдачи исправленного электронного варианта работы и текстовой части работы в бумажном виде.

Студент должен устранить замечания преподавателя по выполненной курсовой работе и сдать ее не позднее срока, установленного преподавателем для окончательной сдачи исправленного электронного варианта работы и текстовой части работы в бумажном виде.

Курсовая работа, сданная студентом после установленного срока, может быть засчитана как выполненная в окончательном виде без возможности устранения замечаний по этой работе.

При использовании дистанционных технологий обучения все вопросы, связанные со сроками выполнения курсовой работы, могут быть решены с использованием электронных средств связи.

### **3.6 Оформление текстовой части курсовой работы**

Текстовая часть курсовой работы должна быть оформлена в соответствии со стандартом ВГУЭС СТП 1.005-2004 (2) и может быть выполнена с использованием текстового редактора Microsoft Word или другого аналогичного текстового редактора.

Текстовая часть курсовой работы должна включать в себя следующие структурные элементы в указанной ниже последовательности:

- титульный лист (обязательный элемент);
- задание (обязательный элемент – в содержание не включается);

- аннотация или реферат (необязательный элемент – в содержание не включается);
- содержание (если текст документа более 10 страниц);
- основная часть (обязательный элемент), включающая в себя введение (необязательный элемент), разделы (обязательные элементы), заключение или выводы (необязательный элемент);
- список использованных источников (обязательный элемент);
- приложения (при необходимости).

Задание – включает в себя формулировку задания на выполнение курсовой работы и может быть уточнено в процессе выполнения работы.

Аннотация или реферат – состоит из краткого изложения содержания курсовой работы.

Содержание (точнее, оглавление) – представляет собой структуру курсовой работы в виде перечня пунктов (заголовков) с указанием для каждого пункта номера страницы, с которой он начинается.

Введение – может состоять из краткого описания предметной области по теме выполняемой работы, основных результатов анализа информации, необходимой для реализации задач курсовой работы, и выбранных способов решения этих задач.

Разделы – могут содержать достаточно подробное описание (раздельное или совместное) предметной области по теме выполняемой работы, результатов анализа информации, необходимой для реализации задач курсовой работы, выбранных способов решения этих задач.

Если выполнение работы предполагает создание проектов, то в разделах могут содержаться инструкции по использованию соответствующих программ, которые могут включать в себя описания функционального назначения и возможностей программ, возможных аварийных ситуаций при их работе и, при необходимости, описания предварительных действий, выполняемых для их запуска, и, опять же при необходимости, описания процесса их установки.

В соответствующие разделы могут также при необходимости включаться описания процессов конструирования и выполнения проектов, фрагменты кодов программ и т.д..

Введение и разделы, а, при необходимости, и другие структурные элементы текстовой части курсовой работы могут содержать формулы, таблицы, рисунки (в том числе схемы, диаграммы) и т.п., которые должны быть соответствующим образом пронумерованы, оформлены желательно в одинаковых стилях и ссылки на которые должны содержаться в текстовой части работы.

Приложения – могут содержать коды программ с необходимыми комментариями, таблицы, рисунки и т.п., а также тексты, которые ввиду

их громоздкости, большого количества или по другим причинам нецелесообразно размещать в других структурных элементах текстовой части курсовой работы.

Приложения должны быть соответствующим образом пронумерованы, и ссылки на них должны содержаться в текстовой части курсовой работы.

Отдельные приложения ввиду их очень большого объема могут не включаться в бумажный вариант текстовой части курсовой работы, а могут существовать только в электронном варианте. Ссылки на такие приложения должны содержаться в бумажном варианте текстовой части курсовой работы. Страницы каждого из таких приложений должны нумероваться, начиная с первой страницы, а в заголовке такого приложения перед строкой с номером этого приложения должна помещаться строка с названием темы курсовой работы.

Использование и оформление структурных элементов текстовой части работы может быть изменено при получении студентом задания на выполнение курсовой работы или в процессе ее выполнения.

### **3.7 Оформление программных модулей**

Программные модули должны быть снабжены достаточно подробными комментариями.

Короткие комментарии можно располагать в строке с комментируемой инструкцией (оператором) языка программирования.

Более подробные комментарии рекомендуется располагать в отдельных строках выше или ниже группы комментируемых инструкций (операторов).

Описания переменных, констант, процедур, функций и т.п. должны сопровождаться комментариями, указывающими на назначение описываемых объектов.

Каждый обработчик событий должен сопровождаться достаточно подробными комментариями, поясняющими выполняемые им действия.

Перед первым обработчиком событий следует также разместить строки с комментариями, содержащими информацию об измененных по сравнению с используемыми по умолчанию значениях свойств компонентов приложения, если эти изменения не очевидны из вида компонентов на форме.

Комментариями должны также сопровождаться группы инструкций (операторов), реализующие наиболее важные и интересные части программы.

Взаимное расположение инструкций (операторов) должно иметь хорошо различимые структуры, соответствующие вложенности этих инструкций и имеющие, например, следующий вид:

```

begin
...
  begin
    ...
    begin
      ...
      end;
    ...
  end;
...
end

```

Безусловные переходы и метки использовать не рекомендуется или, по крайней мере, их использование должно быть сведено к минимуму и сопровождаться исчерпывающими комментариями.

### **3.8 Использование сред программирования при выполнении курсовой работы.\***

До этого в настоящем руководстве не упоминалась конкретная среда программирования.

При выполнении курсовых работ рекомендуется использование среды программирования не выше Delphi7.

Для студентов заочной формы обучения и при использовании дистанционных технологий обучения допускается выполнение курсовых работ с использованием более поздних версий Delphi с обязательным уведомлением об этом путем обязательного указания использованной версии во введении текстовой части курсовой работы (а также, например, указанием этой версии Delphi в имени папки или архивного файла с результатами выполнения курсовой работы).

Для студентов заочной формы обучения и при использовании дистанционных технологий обучения в исключительных случаях при невозможности использования среды программирования Delphi допускается использование среды программирования VBA (с учетом ограниченных возможностей этой среды) с сохранением соответствующих проектов в одном или нескольких файлах Excel. Соответствующим методическим руководством по использованию среды программирования VBA в этом случае может служить (8). При этом во введении текстовой части курсовой работы следует обязательно упомянуть об этом и указать версии использованной среды VBA и программы Excel.

Далее в настоящем руководстве предполагается использование среды программирования Delphi7.

## 3.9 Использование среды программирования Delphi7 при выполнении курсовой работы.\*

Вопросы использования среды программирования Delphi7 подробно освещены в (3-7), а также могут рассматриваться во время проведения лекционных и лабораторных занятий по дисциплине "Высокоуровневые методы информатики и программирования".

В настоящем руководстве рассмотрены лишь некоторые важные вопросы использования среды программирования Delphi7, незнание ответов на которые может привести к невозможности правильного и эффективного ее использования на начальном этапе работы с ней.

### 3.9.1 Запуск программы Delphi7

Для запуска программы Delphi7 можно выполнить команды:

"Пуск => Программы => Borland Delphi 7 => Delphi 7"

(ЗАПУСК ПРОГРАММЫ МОЖЕТ ПРОИСХОДИТЬ ОЧЕНЬ ДОЛГО).

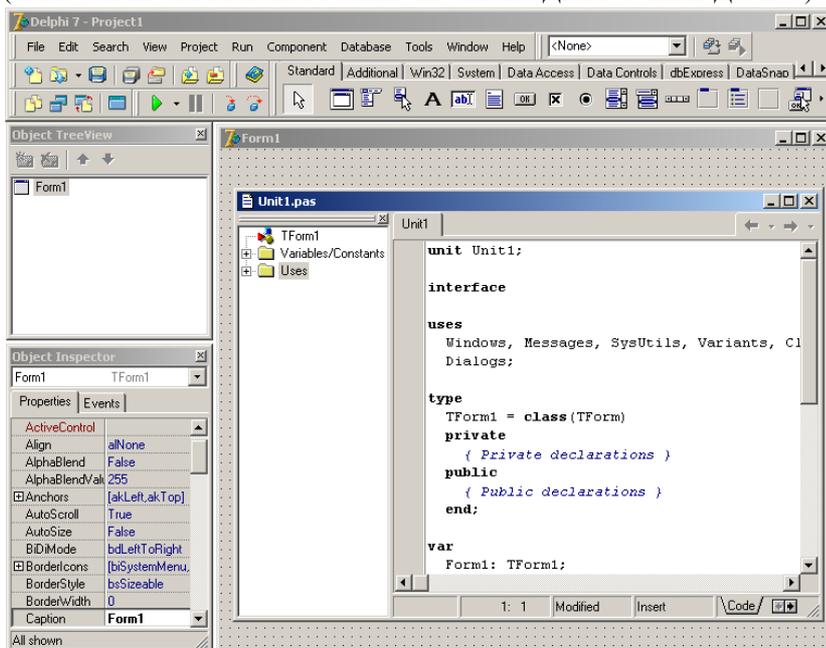


Рис. 2

После запуска на экране обычно появляются окна среды программирования Delphi7 и окна формы и модуля создаваемого по умолчанию проекта. На рисунке 2 показано окно среды программирования Delphi7 после запуска программы и активизации окна модуля Unit1.pas с видимым окном формы Form1. В заголовке окна

Delphi 7 находится имя создаваемого проекта (в данном случае – Project1).

Для создания нового проекта можно также выполнить команды меню окна Delphi 7 "File => New => Application" и, после возможного выполнения действий, связанных с сохранением файлов ранее открытого проекта, окно среды программирования приобретет вид, аналогичный виду на рисунке 2.

### **3.9.2 Последовательность основных действий с проектом**

Действия с проектом обычно начинаются с "перетаскивания" с помощью указателя "мыши" значков компонентов в окно формы с вкладок Палитры компонентов, расположенной в правой части окна Delphi 7.

Во время "перетаскивания" значков в окно формы или после него можно изменить с помощью указателя "мыши" расположение этих значков в окне формы и их размеры.

При необходимости, используя вкладку Properties окна Object Inspector, можно изменить значения свойств компонентов, значки которых расположены в окне формы, а также изменить значения свойств самой формы.

Используя вкладку Events окна Object Inspector, можно создать в окне модуля проекта обработчики событий (точнее, процедуры обработчиков событий) для компонентов, значки которых расположены в окне формы, а также обработчики событий для самой формы, или выбрать для использования уже имеющиеся обработчики событий.

После создания обработчиков событий и ввода необходимых инструкций (операторов) и других данных в окнах модулей проекта следует сохранить файлы проекта (рекомендуется в специально отведенной для этого проекта папке).

Сохранение файлов проекта целесообразно выполнить уже в самом начале работы с проектом (это можно даже сделать сразу после создания проекта), а в процессе работы периодически сохранять его файлы.

Сохранение файлов проекта можно выполнять с использованием соответствующих команд из меню команды "File" в окне Delphi 7 ("Save As...", "Save Project As...", "Save").

Запуск созданного приложения в среде программирования Delphi7 можно выполнить, если нажать на кнопку "Run" на панели инструментов "Debug", или нажать клавишу F9 на клавиатуре, или выполнить команды меню окна Delphi 7 "Run => Run" (после запуска в папке с сохраненными файлами проекта появится также выполняемый файл приложения с расширением ".exe").

Для запуска созданного приложения независимо от среды программирования можно, например, открыть папку, в которой были сохранены файлы созданного проекта и запустить выполняемый файл (с расширением "exe").

При необходимости отладки созданного приложения следует использовать соответствующие команды из меню команды "Run" в окне Delphi 7 или соответствующие им клавиши и сочетания клавиш (в частности, клавишу F7 и соответствующую ей команду "Trace Into").

При невозможности закрытия окна приложения, запущенного в среде программирования Delphi7, следует выполнить команды меню окна Delphi 7 "Run => Program Reset".

### **3.9.3 Особенности доступа к свойствам компонентов и рекомендации по использованию свойств**

Для доступа к свойствам компонента, значок которого расположен на форме (или для доступа к свойствам формы), достаточно привести указатель "мыши" на этот значок (для формы – на участок формы, не занятый значками) и выполнить один щелчок левой кнопкой "мыши", после чего при открытой вкладке Properties окна Object Inspector, будет получен доступ к свойствам компонента (или формы).

Кроме того, для доступа к свойствам можно использовать значки компонентов и формы в окне Object TreeView или строки раскрывающегося списка с именами компонентов в верхней части окна Object Inspector.

Не рекомендуется оставлять неизменными предлагаемые по умолчанию для компонентов и формы значения свойства Caption, а также значения тех других свойств, которые (значения) могут быть видны в окне выполняемого приложения (такие, например, как Form1, Button1, Edit1 и т.п.).

### **3.9.4 Особенности использования обработчиков событий и рекомендации по их использованию**

Для создания в окне модуля проекта обработчика заданного события для какого-либо компонента (или формы) можно тем или иным способом выбрать этот компонент (или форму) по аналогии со способами выбора компонента для изменения его свойств и затем на вкладке Events окна Object Inspector найти строку с заданным событием, в этой же строке привести указатель "мыши" на область раскрывающегося списка правее названия события и выполнить двойной щелчок левой кнопкой "мыши".

При этом в окне модуля будет создана "пустая" процедура обработчика именно заданного события.

Для того чтобы использовать для обработки заданного события уже имеющийся обработчик событий, достаточно на вкладке Events окна Object Inspector найти строку с заданным событием и в этой же строке в области раскрывающегося списка правее названия события выбрать один из имеющихся обработчиков событий.

Для быстрого доступа к обработчикам событий компонента, значок которого расположен на форме (или к обработчикам событий формы), достаточно навести указатель "мыши" на этот значок (для формы – на участок формы, не занятый значками) и выполнить двойной щелчок левой кнопкой "мыши".

В процедуре обработчика события рекомендуется вводить инструкции (операторы) только между строками с **begin** и **end**; Настоятельно не рекомендуется без достаточного опыта работы со средой программирования "трогать" чем-либо, кроме комментариев, эти строки с **begin** и **end**; строку с именем и параметрами созданной процедуры, а также последнюю строку модуля, содержащую **end**.

### 3.9.5 Особенности сохранения файлов проекта

При сохранении файлов проекта для сохранения этих файлов по умолчанию обычно предлагается стандартная папка с именем Projects, создаваемая при установке системы программирования Delphi7 на компьютере.

Категорически не рекомендуется использовать эту стандартную папку, предлагаемую средой программирования при сохранении файлов проекта, так как некорректное сохранение файлов проекта в этих папках (особенно в сочетании с использованием предлагаемых по умолчанию средой программирования имен файлов, таких, например, как Project1 и Unit1), может привести к замене отдельных файлов проекта файлами другого проекта и, как следствие, невозможности в дальнейшем правильного открытия проекта.

При сохранении файлов проекта следует создать папку (это можно сделать прямо в соответствующем диалоговом окне при выполнении сохранения файла), предназначенную для сохранения только файлов этого проекта и затем использовать только эту папку для сохранения файлов этого проекта.

В среде программирования Delphi файл проекта и файл модуля не могут иметь одинаковые имена, несмотря на то, что эти файлы имеют разные расширения (соответственно, `dpr` и `pas`).

Особенно внимательным следует быть при попытке открытия или создания нового проекта до полного закрытия всех файлов ранее открытого проекта. В такой ситуации, при появлении окон с вопросами о необходимости сохранения файлов ранее открытого проекта, следует внимательно прочитать тексты вопросов в этих окнах и правильно

выполнить действия, связанные с сохранением или несохранением указанных файлов.

### 3.9.6 Настройка и изменение состава видимых окон среды программирования

Настройка панелей инструментов в окне среды программирования Delphi7 выполняется с использованием вкладок окна "Customize" (см. рисунки 3 и 4) которое появляется на экране после выполнения команды "Customize..." из контекстного меню для области панелей инструментов или для области строки меню окна Delphi7 или команд меню окна Delphi7: "View => Toolbars=> Customize...". Указанная настройка выполняется по аналогии с настройкой панелей инструментов в окнах программ пакета Microsoft Office.

Другие настройки вида и параметров среды программирования могут быть выполнены с использованием команд меню окна Delphi7 "View " или "Tools".

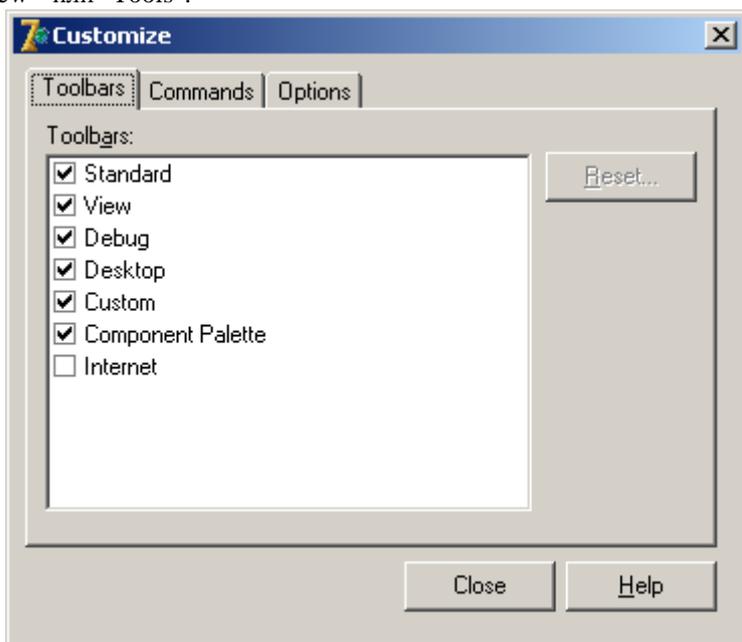


Рис. 3

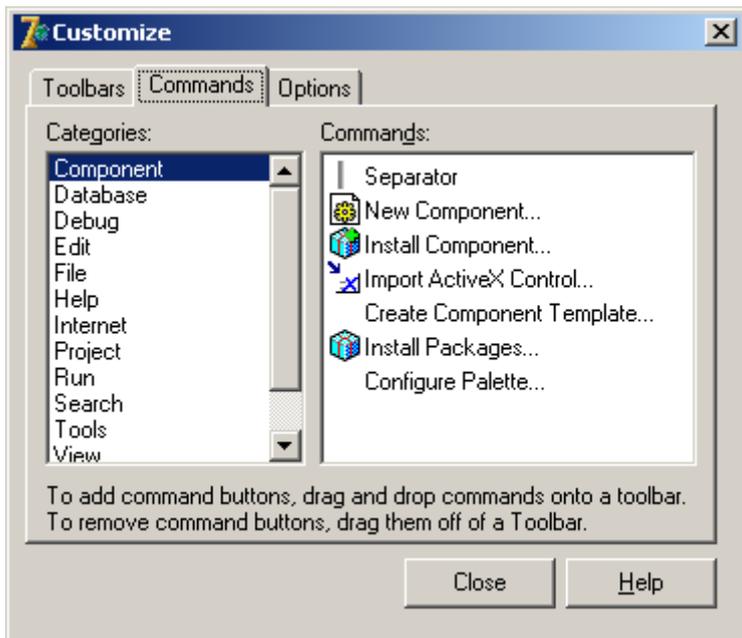


Рис. 4

### 3.9.7 Использование консольных приложений

Консольные приложения можно использовать для быстрой отладки алгоритмов, в том числе с непосредственным вводом и выводом числовых значений (с помощью процедур `read`, `readln` и `write`, `writeln`) без использования подпрограмм для преобразования текстовых величин в числовые и обратно.

Для перехода к созданию консольного приложения в окне Delphi7 можно выполнить команды меню "File => New => Other..." (или нажать на панели инструментов "Standard" кнопку "New items") и затем в появившемся окне "New Items" на вкладке "New" выбрать значок "Console Application" (см. рисунок 5).

После выбора этого значка на экране должно появиться окно проекта создаваемого консольного приложения (см. рисунок 6).

После запуска созданного консольного приложения на экране появляется окно, обычно с черным фоном, в котором можно вводить данные, используемые этим приложением, и выводить результаты вычислений.

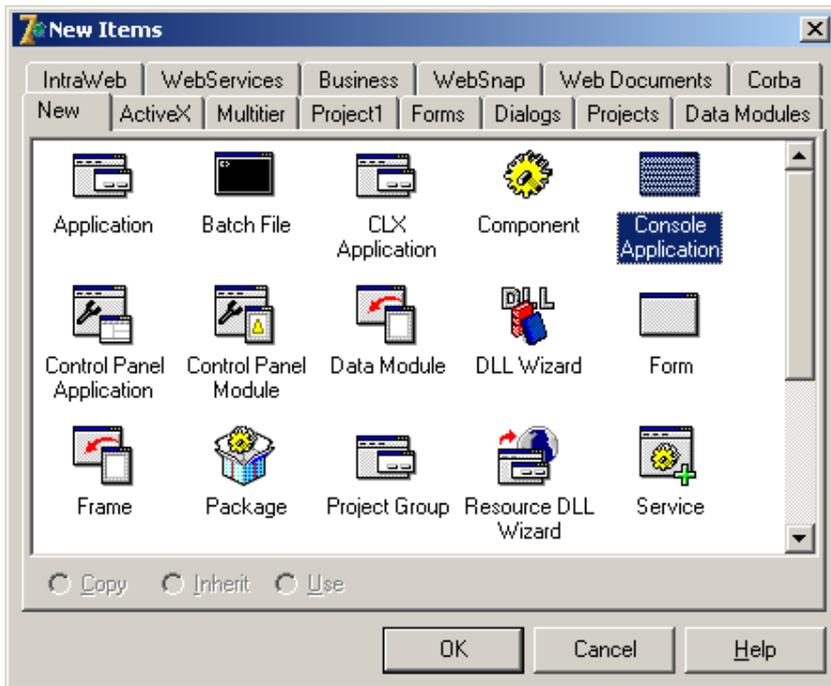


Рис.5

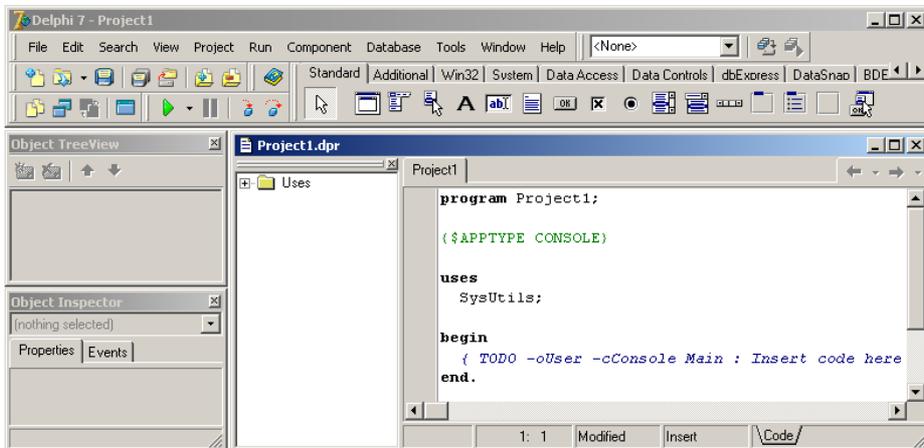


Рис.6

### 3.9.8 Пример задания на выполнение курсовой работы с фрагментами текстовой части курсовой работы и с приложением программного кода

Ниже приведен пример весьма упрощенного задания на выполнение курсовой работы с отдельными фрагментами текстовой части соответствующей курсовой работы и представлено приложение к текстовой части работы с программными кодами, снабженными подробными комментариями.

#### ЗАДАНИЕ

Создать с использованием среды программирования Delphi программное приложение, реализующее решение квадратного уравнения с действительными коэффициентами.

В верхней части окна приложения должны находиться три поля Edit для ввода коэффициентов, рядом с ними – поясняющие надписи, которые вместе с введенными в поля значениями должны изображать общепринятую запись квадратного уравнения вида  $ax^2+bx+c=0$ .

В средней части окна должны находиться две кнопки: "Решение" и "Очистить".

При нажатии на кнопку "Решение" нечисловые значения в полях ввода должны обнуляться и на месте этих нечисловых значений должны появляться нули, после чего должно выполняться решение уравнения, коэффициенты которого находятся в полях в верхней части окна, и результаты решения должны быть выведены в нижней части окна в виде надписей в общепринятом виде для записи как действительных, так и комплексных корней квадратного уравнения, а также частных решений, когда уравнение вырождается в линейное или когда коэффициенты  $a$  и  $b$  одновременно равны нулю (алгоритм решения уравнения, соответственно, должен предусматривать все эти случаи).

При нажатии на кнопку "Очистить" поля ввода и места вывода результатов решения должны становиться пустыми.

#### ОКНО ПРИЛОЖЕНИЯ

На рисунке 7 представлена форма созданного программного приложения на стадии проектирования.

В левом нижнем углу формы видны четыре выделенные сейчас метки Label1,2,3 и 4. Значения по умолчанию свойства Caption меток Label1 и 2 изменены на пустые значения для очистки места вывода значений корней уравнения при запуске приложения. Значение свойства Caption метки Label3 равно 1 – индексу первого корня уравнения, значение этого же свойства метки Label4 равно 2 – индексу второго корня уравнения. Значения по умолчанию

свойства Visible меток Label3 и 4 изменены на false для очистки места вывода значений корней уравнения при запуске приложения.

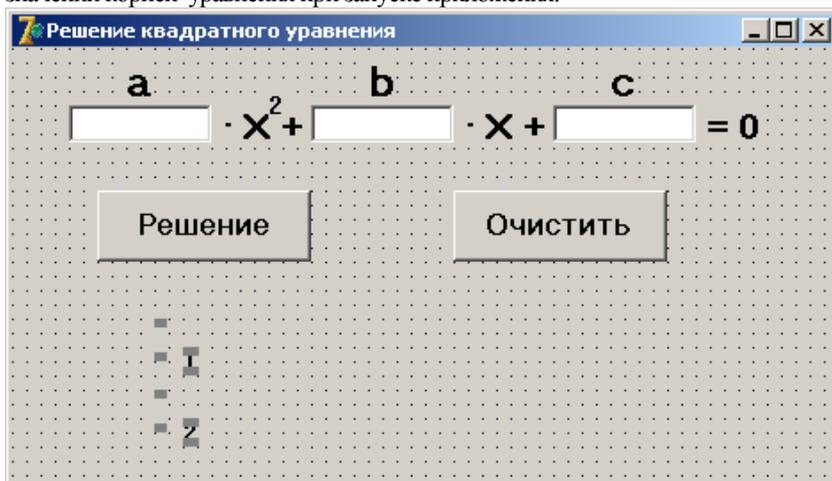


Рис.7

Из этих меток левая верхняя метка Label1 используется, в случае двух корней уравнения, для вывода надписи  $x=$  и далее значения корня  $x_1$ , в остальных случаях – для вывода надписи со значением решения или надписи об отсутствии решения.

Правая сверху метка Label3 используется только в случае двух корней уравнения для вывода 1 – индекса первого корня (в этом случае ее свойство Visible имеет значение true), в остальных случаях ее свойство Visible имеет значение false.

Левая снизу метка Label2 используется только в случае двух корней уравнения для вывода надписи  $x=$  и далее значения корня  $x_2$  (в остальных случаях ее свойство Caption имеет пустое значение).

Правая снизу метка Label4 – используется только в случае двух корней уравнения для вывода 2 – индекса второго корня (в этом случае ее свойство Visible имеет значение true), в остальных случаях ее свойство Visible имеет значение false.

Остальные метки в верхней части окна формы являются статичными и не меняют своих значений.

Вид окна приложения с различными результатами его работы показан на рисунках 8 и 9.

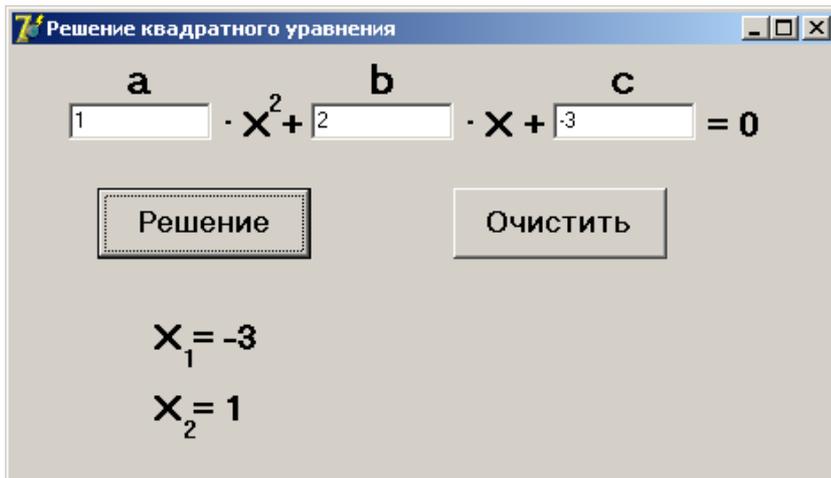


Рис.8

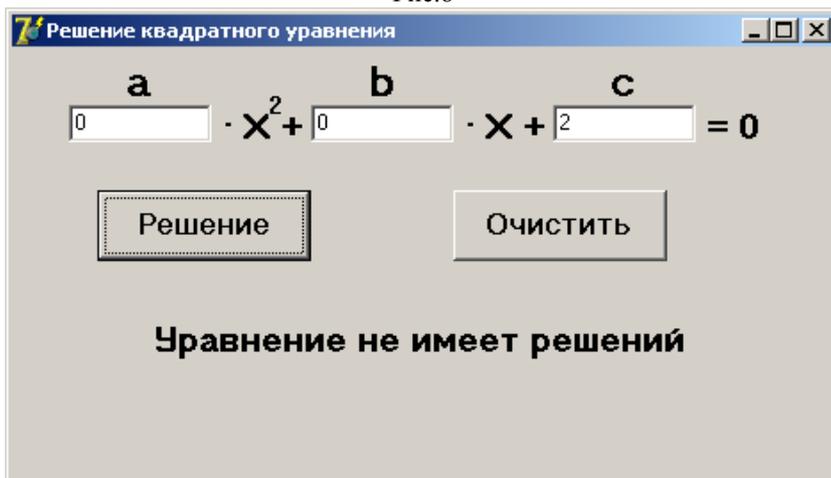


Рис.9

## ПРИЛОЖЕНИЕ.

Программный код приложения

```
unit Kvr1;
```

```
//Решение квадратного уравнения
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,  
Forms, Dialogs, StdCtrls;
```

**type**

```
TForm1 = class(TForm)
  Edit1: TEdit;
  Edit2: TEdit;
  Edit3: TEdit;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Button1: TButton;
  Button2: TButton;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  Label9: TLabel;
  Label10: TLabel;
  Label11: TLabel;
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
```

**private**

```
  { Private declarations }
```

**public**

```
  { Public declarations }
```

```
end;
```

**var**

```
Form1: TForm1;
a,b,c,D: real; // Коэффициенты и дискриминант квадратного
               // уравнения
```

**implementation**

```
{ $R *.dfm }
```

```
{ Для очистки места вывода значений корней уравнения
  при запуске приложения значения свойства Caption
  меток Label1,2 изменены на пустые значения,
  значения свойства Visible меток Label3,4
  изменены на false }
```

```
// Обработка нажатия на кнопку "Решение"
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
Label M; { Описание программной метки для перехода goto M
           в конец этого обработчика событий.
```

*В данном случае использование программной метки и инструкции goto M делает алгоритм более наглядным, чем использование нескольких вложенных друг в друга условных переходов без программной метки }*

**begin**

```

Label1.Caption:="; // Присвоение свойству Caption меток Label1,2
Label2.Caption:="; // пустого значения,
Label3.Visible:=false; // свойству Visible меток Label3,4 значения false
Label4.Visible:=false; // для очистки места вывода значений корней
// уравнения
a:=StrToFloatDef(Edit1.Text,0); // Присвоение переменным a, b, c
// введенных
b:=StrToFloatDef(Edit2.Text,0); // значений коэффициентов уравнения
c:=StrToFloatDef(Edit3.Text,0); // с обнулением нечисловых значений
Edit1.Text:=FloatToStr(a); // Переприсвоение введенных значений
Edit2.Text:=FloatToStr(b); // полям ввода на случай обнуления
// введенных
Edit3.Text:=FloatToStr(c); // нечисловых значений
if (a=0)and(b=0)and(c=0) // Вырожденное уравнение вида 0=0
  then begin Label1.Caption:='X - любое число'; goto M end;
if (a=0)and(b=0)and(c<>0) // Вырожденное уравнение вида c=0,
  // где c не равно 0
  then begin Label1.Caption:='Уравнение не имеет решений'; goto M
  end;
if (a=0)and(b<>0) // a=0, уравнение вырождается в линейное вида
  // bx+c=0
  then begin Label1.Caption:=FloatToStr(-c/b); goto M end; // Один
  // корень
D:=Sqr(b)-4*a*c; // Вычисление дискриминанта
if D>=0 // Два действительных корня
  then
  begin
  Label1.Caption:='X = '+FloatToStr((-b-Sqrt(D))/2/a); // Меньший
  // корень
  Label3.Visible:=true; // Виден индекс первого корня
  Label2.Caption:='X = '+FloatToStr((-b+Sqrt(D))/2/a); // Больший
  // корень
  Label4.Visible:=true; // Виден индекс второго корня
  goto M
  end;
  end;
// Дискриминант меньше 0: два комплексных корня,
// i - мнимая единица

```

```

Label1.Caption:='X = '+FloatToStr(-b/2/a)+' - i'+FloatToStr(Sqrt(-D)/2/a);
Label3.Visible:=true; // Виден индекс первого корня
Label2.Caption:='X = '+FloatToStr(-b/2/a)+' + i'+FloatToStr(Sqrt(-D)/2/a);
Label4.Visible:=true; // Виден индекс второго корня
M: // Пустая инструкция с программной меткой в конце обработчика
    // событий
end;

// Обработка нажатия на кнопку "Очистить"
procedure TForm1.Button2Click(Sender: TObject);
begin
    Edit1.Text:=""; // Присвоение полям ввода
    Edit2.Text:=""; // пустых значений для очистки
    Edit3.Text:=""; // мест ввода значений коэффициентов уравнения
    Label1.Caption:=""; // Присвоение свойству Caption меток Label1,2
    Label2.Caption:=""; // пустого значения,
    Label3.Visible:=false; // свойству Visible меток Label3,4 значения false
    Label4.Visible:=false; // для очистки места вывода значений корней
    // уравнения
end;

end.

```

## 4. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

### 4.1 Основная литература

1. Учебная программа по дисциплине "Высокоуровневые методы информатики и программирования / Сост. В.И. Люлько. – Владивосток: Изд-во ВГУЭС, 2005. – 18 с.
2. СТП 1.005-2004. "Система вузовской учебной документации. Общие требования к оформлению текстовой части дипломных, курсовых работ (проектов), рефератов, контрольных работ, отчетов по практикам, лабораторным работам. Структура и правила оформления. Стандарты Владивостокского государственного университета экономики и сервиса" / Авт.-сост. Н.И. Попова. – Владивосток: Изд-во ВГУЭС, 2004. – 44 с.
3. Информационные системы /Петров В.Н. – СПб.: Питер, 2003. – 688 с.
4. Delphi. Программирование на языке высокого уровня: Учебник для вузов / В. В. Фаронов. – СПб.: Питер, 2005. – 640 с.
5. Культин Н.Б. Delphi в задачах и примерах. – СПб.: БХВ-Петербург, 2004. – 288 с.

#### **4.2 Дополнительная литература**

6. Архангельский А.Я. Программирование в Delphi 7 – М.: ООО "Бином-Пресс" 2003 г. – 1152 с.
7. Архангельский А.Я. Delphi 7. Справочное пособие, М.: ООО "Бином-Пресс" 2003 г. – 1024 с.
8. Информатика и программирование: Руководство по выполнению курсовых работ / Сост. С.Л. Бедрина, О.Б. Богданова. – Владивосток: Изд-во ВГУЭС, 2005. – 64 с.

# ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	3
1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ .....	4
1.1 Цель и задачи курсовой работы .....	4
1.2 Требования к выполнению курсовой работы .....	4
1.3 Объем и содержание курсовой работы .....	5
1.4 Техническое и программное обеспечение дисциплины .....	5
1.5 Виды контроля знаний студентов и их отчетности по курсовой работе .....	5
2. ТЕМАТИКА КУРСОВЫХ РАБОТ .....	6
3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВЫХ РАБОТ .....	7
3.1 Выбор темы курсовой работы и получение задания на ее выполнение .....	7
3.2 Порядок размещения и рекомендуемые имена папок и файлов с результатами работы .....	8
3.3 Выполнение курсовой работы .....	9
3.4 Представление результатов выполнения курсовой работы .....	10
3.5 Сроки выполнения курсовой работы .....	11
3.6 Оформление текстовой части курсовой работы .....	11
3.7 Оформление программных модулей .....	13
3.8 Использование сред программирования при выполнении курсовой работы.* .....	14
3.9 Использование среды программирования Delphi7 при выполнении курсовой работы.* .....	15
3.9.1 Запуск программы Delphi7 .....	15
3.9.2 Последовательность основных действий с проектом .....	16
3.9.3 Особенности доступа к свойствам компонентов и рекомендации по использованию свойств .....	17
3.9.4 Особенности использования обработчиков событий и рекомендации по их использованию .....	17
3.9.5 Особенности сохранения файлов проекта .....	18
3.9.6 Настройка и изменение состава видимых окон среды программирования .....	19
3.9.7 Использование консольных приложений .....	20
3.9.8 Пример задания на выполнение курсовой работы с фрагментами текстовой части курсовой работы и с приложением программного кода .....	22

4. СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ .....	27
4.1 Основная литература .....	27
4.2 Дополнительная литература .....	28

Учебно-методическое издание

**Виктор Иванович Люлько**

РУКОВОДСТВО ПО ВЫПОЛНЕНИЮ КУРСОВЫХ РАБОТ  
ПО ДИСЦИПЛИНЕ  
«ВЫСОКОУРОВНЕВЫЕ МЕТОДЫ  
ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ»

Редактор  
Корректор  
Компьютерная верстка

Лицензия на издательскую деятельность ЛР № 021014 от 03.11.1995

99039

Подписано в печать Формат 60×84 1/16.  
Бумага типографская. Печать офсетная. Усл. печ. л. 2,0.  
Тираж 100 экз. Заказ

---

Издательство Владивостокского государственного университета  
экономики и сервиса

690600, Владивосток, ул. Гоголя, 41

Отпечатано в лаборатории множительной техники ВГУЭиС  
690600, Владивосток, ул. Державина, 57